

PATENT

Attorney Docket No. 07781.0042

CUSTOMER NUMBER 22,852

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re U.S. national phase of)
PCT/EP01/06701)
Inventors: Juergen HEYMANN et al.) Group Art Unit: Unassigned
Serial No.: Not Yet Assigned) Examiner: Unassigned
Filed: February 13, 2002)
For: COMMUNICATION BETWEEN)
CLIENT AND SERVER)
COMPUTERS VIA HTTP,)
METHOD, COMPUTER)
PROGRAM PRODUCT AND)
SYSTEM)

**Assistant Commissioner for Patents
Washington, DC 20231**

BOX: PCT

Sir:

PRELIMINARY AMENDMENT

Prior to examination, please amend the above-identified application as follows:

IN THE SPECIFICATION:

Please amend the specification as follows:

Page 1, before the first paragraph insert the following new paragraph:

--This application claims the benefit of provisional application number
60/211,629, filed June 14, 2000 the content of which is incorporated herein by
reference. --

FINNEGAN
HENDERSON
FARABOW
GARRETT &
DUNNER LLP

1300 I Street, NW
Washington, DC 20005
202.408.4000
Fax 202.408.4400
www.finnegan.com

REMARKS

The examiner is respectfully requested to consider the above preliminary amendment prior to examination of the application.

If there are any other fees due in connection with the filing of this amendment, please charge the fees to Deposit Account No. 06-0916. If a fee is required for an extension of time under 37 C.F.R. § 1.136 not accounted for above, such an extension is requested and the fee should also be charged to our deposit account.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: February 13, 2002

By: 

Ernest F. Chapman
Reg. No. 25,961

EFC/FPD/peg

FINNEGAN
HENDERSON
FARABOW
GARRETT &
DUNNER LLP

1300 I Street, NW
Washington, DC 20005
202.408.4000
Fax 202.408.4400
www.finnegan.com



JC02 Rec'd PCT/PTO 1 3 MAY 2002 #3

PATENT
Customer No. 22,852
Attorney Docket No. 07781.0042

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:)
)
J. Heymann et al.) Group Art Unit: Unassigned
)
Application No.: 10/049,522) Examiner: Unassigned
)
Filed: February 13, 2002)
)
For: COMMUNICATION BETWEEN)
CLIENT AND SERVER)
COMPUTERS VIA HTTP,)
METHOD, COMPUTER)
PROGRAM PRODUCT AND)
SYSTEM)

Assistant Commissioner for Patents
Washington, DC 20231

Sir:

SUPPLEMENTAL PRELIMINARY AMENDMENT

Further to the Preliminary Amendment of February 13, 2002, and prior to the first official examination, please amend the present application as follows:

IN THE SPECIFICATION:

Please amend the specification as follows:

Page 1 paragraph 3

Typically, the user operates a personal computer (referred to as "client computer") that has to communicate with a remote computer (referred to as "server computer"). The client computer has communication software to communicate with the

server computer and the server computer has application software to execute a business application.

Page 2 first full paragraph

Using specialized communication software at the client computer is inconvenient. Besides the time that is required to install it, installing the communication software might require the payment of license fees. Also, regular and costly updates are required. There is a tendency to communicate with standard "off-the-shelf" software such as Internet browsers. Browsers are installed in almost every personal computer. The hypertext transfer protocol (HTTP) became the standard communication protocol in the Internet. However, HTTP does not automatically notify the server computer about a session termination by the client computer.

Page 3 paragraph 2

According to an embodiment of the present invention, a method is provided for communication between a client computer and a server computer that both use HTTP. The client computer uses an HTTP-browser. The method comprises the following steps:

Page 4 paragraph 1

It is an advantage that the close instruction in the browser virtually couples the client computer to the server computer. In the event of unloading, the server computer is notified and is able to release the resource. Further, the browser may be a standard browser that interprets the close instruction but that does not need to be modified.

Page 4 paragraph 3 (bridging to page 5)

According to another embodiment of the present invention, a computer program product is provided for HTTP-communication between a client computer and a server computer, wherein the client computer has a browser. The computer program product has program code portions that cause a client processor in the client computer and a server processor in the server computer to control the communication. The computer program product comprises: code portions that cause the client processor to send a first request to the server computer; code portions that - upon receiving the first request by the server computer - cause the server processor to (i) allocate a resource at the server computer, the resource including an identifier, and (ii) return a predetermined close instruction to the browser, the close instruction carrying the identifier; code portions that - upon unloading the close instruction from the browser of the client computer - cause the client processor to send a second request to the server computer, the second request carrying the identifier and indicating to de-allocate the resource; and code portions that - upon receiving the second request from the client computer - cause the server processor to de-allocate the resource.

Page 5 first full paragraph

Preferably, the code portions cause the client processor to provide such a close instruction that the browser provides a first frame to present the close instruction in a first frame and provides a second frame to present content pages that the client computer receives from the server computer. Preferably, the code portions cause the client processor to provide a close instruction such that the caching of selected content pages by the browser is prevented. Preferably, the code portions cause the client

processor to provide such a close instruction so that the client computer sends the second request to a predetermined address of the server computer.

Page 5 paragraph 2

According to still another embodiment of the present invention, computer readable media is provided that separately stores the program code portions causing the client processor and the server processor to operate.

Page 6 paragraph 1

According to another embodiment of the present invention, a computer system is provided that includes a client computer and a server computer, both computers using HTTP for communication and the client computer using an HTTP-browser. The client computer sends a first request to the server computer; the server computer (upon receiving the first request) (i) allocates a resource (resource including an identifier), and (ii) returns a predetermined close instruction to the browser of the client computer (the close instruction carrying the identifier); the client computer (upon unloading the close instruction from the browser) sends a second request to the server computer (the second request carrying the identifier and indicating to de-allocate the resource); and the server computer (upon receiving the second request from the client computer) de-allocates the resource.

Page 6 last paragraph (bridging to page 7)

According to yet another embodiment of the present invention, a method is provided for communication between a client computer and a server computer. Both computers use HTTP and the client computer uses an HTTP-browser. The client computer sends a request to the server computer. Upon receiving the request, the

server computer: allocates a resource at the server computer (the resource including an identifier and a time-out period), returns a close instruction to the client computer (the close instruction including the time-out period and the identifier), measures the time during which communication between the client and server computers is idle, and de-allocates the resource when the measured time reaches the time-out period. Upon receiving the close instruction, the client computer measures the time during which the communication between the client computer and the server computer is idle, and displays a warning to the user if the measured time reaches a predetermined fraction of the time-out period.

Page 7 paragraph 2 (bridging to page 8)

In accordance with another embodiment of the present invention, a computer program product is provided for controlling HTTP-communication between a client computer and a server computer, wherein the client computer has a browser. The computer program product has a client program portion to control a client processor and a server program portion to control a server processor. The program is characterized in that the client program product portion causes the client processor to send a request from the client computer to the server computer; upon receiving the request by the server computer, the server program portion causes the server processor to allocate a resource at the server computer (resource with identifier and time-out period (T)), to return a close instruction to the client computer (close instruction with time-out period (T) and identifier), to measure the time (t) during which communication between the client computer and the server computer is idle, and to de-allocate the resource when the measured time (t) reaches the time-out period (T); and upon receiving the close

instruction by the client computer, the client program portion causes the client processor to measure the time (t) during which the communication between the client computer and the server computer is idle, and to display a warning to the user if the measured time (t) reaches a predetermined fraction (T/X) of the time-out period (T).

Page 8 first full paragraph

According to still another embodiment of the present invention, a method is provided for communication between a client computer and a server computer, (HTTP, client computer with HTTP-browser). The method comprises: sending a first request from the client computer to the server computer; allocating a resource at the server computer, the resource including an identifier; returning a predetermined response page to the browser, the response page carrying the identifier and carrying browser instructions; as instructed by the response page, periodically sending the second requests by the browser to the server computer, the second requests by the browser to the server computer, the second requests carrying the identifier; and at the server computer, periodically checking the arrival of the second requests with the identifier from the client computer and de-allocating the resource when a predetermined time period (T) has lapsed since the last arrival.

Page 9 last paragraph (bridging to page 10)

Computers 900-902 are coupled via inter-computer network 990. Computer 900 comprises 910, memory 920, bus 930, and, optionally, input device 940 and output device 950 (I/O devices, user interface 960). As illustrated, features of the invention may be implemented by computer program product 100 (CPP), program carrier 970 and program signal 980, collectively "program".

Page 14 paragraph 2

FIG. 2 illustrates a simplified block diagram of system 999 including a client computer 900 and a server computer 901 that communicate with each other via network 990 (branch 990-1). Both computers 900 and 901 use the hypertext transfer protocol (HTTP), and client computer 900 further uses HTTP-browser 210. Browser 210 is the program to locate content pages (e.g., in computer 901) and to display the content pages (presentation 21, cf. FIG. 4). Preferably, browser 210 presents graphics as well as text. Preferably, browser 210 is a Netscape Navigator or a Microsoft Internet Explorer.

Page 15 paragraph 1

During normal operation, browser 210 requests content pages from server computer 901, and server computer 901 responds with content (e.g., HTML pages). Browser 210 then causes display 950 to show content pages to the user. An exemplary content page 335 is symbolized by an exclamation mark.

Page 15 paragraph 2

Similar elements such as display, memory, processor, etc. for the other computers are not illustrated for simplicity. Optionally, client computer 900 also communicates with computers 902 and 903 via branches 990-2 and 990-3, respectively. Further, computers 902 and 903 execute applications 402 and 403, respectively. Optionally, the content pages are generated by application computer 902. Likewise, browser 210 requests pages from application computer 902 and application computer 902 responds with application pages.

Page 15 paragraph 3

Preferably, application 403 assists the user to identify application 401 and 402 out of a plurality of applications that are available in the overall network. Such assistance applications are commercially available from SAP Aktiengesellschaft, Walldorf (Baden), Germany under the name "workplace". For convenience, computer 903 is therefore referred to as "workplace computer". Further illustrated elements are: resource 340 (used temporarily), requests 230, 240, instruction 360 and ID 350 (transmitted via the network). The functions of these elements are explained below.

Page 15 paragraph 4

FIG. 3 illustrates a simplified flow chart diagram of method 500 of the present invention. Method 500 is a method of communication between client computer 900 and server computer 901. Both computers 900, 901 use the hypertext transfer protocol (HTTP) and client computer 900 uses HTTP-browser 210.

Page 17 paragraph first full paragraph

While the present invention is described in connection with a single resource 340, a single session and a single identifier 350, persons of skill in the art are able to implement two or more sessions in parallel.

Page 18 paragraph first full paragraph

It is also possible - although not required for the present invention - that the user terminates the session explicitly, for example, by operating a functional button such as an "abort session" button (cf. 217, 218 or 219 in FIG. 4 or similar ones).

Page 19 paragraph 2

FIG. 4 illustrates display 950 of client computer 900 with which HTTP-browser 210 (cf. FIG. 2) generates browser presentation 211. Browser presentations are well

known in the art. In the example of FIG. 4, presentation 211 has back button 213, address field 214, close button 219, warning 205 and a display area for showing frames.

Page 19 paragraph 5 (bridging to page 20)

Splitting the display screen into frames 215, 216 is convenient for the user. Frame 215 informs that session management (cf. method 500) is active, for example, by informing that the user can now access application 401 (e.g., "APPLICATION READY").

Page 22 paragraph 2

Code section 2 defines a global string to store second request 240 (referred to as "termination URL") for the session in child frame 216. In the example, request 240 does not have any other content. It is sufficient to target second request 240 by identifier 350 to an address in server computer 901 that is reserved for de-allocating resource 340 (for example, <http://network-990.server-computer-901.application-401.resoure-340>).

Page 22 last paragraph (bridging to page 23)

"This is first frame 215. Client computer 900 with browser 210 had been sending 520 first request 230 (e.g., by URL <http://network-990/server-computer-901/application-401>) to server computer 901. Upon receiving 530 first request 230, server computer 901 had been allocating 531 resource 340 with identifier 350 and had been returning 532 predetermined close instruction 360 in the form of the present HTML-document "instruction-360.htm" with identifier "340". Close instruction 360 carries identifier 350 ("340"). HTML-document "instruction-360.htm" comprises a termination command in the program section 6 with "onunload". The session has now started."

Page 23 paragraph 5

The advanced session management feature (c) "time" is explained below.

Page 25 last paragraph

The advanced session management feature (d) "cache prevention" is explained further below.

Page 25 paragraph 4

As long as application 401 in server computer 901 is a read-only type application and content pages 335 substantially remain unchanged, displaying cached pages is convenient. In some instances, especially for business applications, intermediate results stored in resource 340 at server computer 901 diverge from temporarily cached content pages 335(t-1) on client computer 900. This is no longer convenient and might cause serious problems.

Page 26 paragraph 3

The following describes an optional implementation of distributed session management.

Page 26 last paragraph (bridging to page 27)

FIG. 9 illustrates a simplified diagram of computer system 999 for further optional method implementations. Illustrated are computers 900, 901 and 903. Applications are distinguished into main (M) and starter (S) applications, addressed by equal domain names. By providing close instructions, the starter applications (S) provide session management functionality according to the present invention (methods 500 and 600). The example of FIG. 9 is simplified to application 401. Persons of skill in the art can use the same scheme for multiple applications (with different domains) in parallel and independent from each other.

Page 28 first full paragraph

The following describes a still further optional embodiment.

IN THE CLAIMS:

Please cancel claims 1-19 without prejudice and enter new claims 20-38 as follows:

---20. A method for communication between a client computer and a server computer, wherein both the client computer and the server computer use the hypertext transfer protocol (HTTP) and the client computer uses an HTTP-browser, the method comprising:

 sending a first request from the client computer to the server computer;

 upon receiving the first request, the server computer (i) allocating a resource at the server computer, the resource including an identifier, and (ii) returning a predetermined close instruction to the browser, the close instruction carrying the identifier;

 upon unloading the close instruction from the browser of the client computer, sending a second request from the client computer to the server computer, the second request carrying the identifier and indicating to de-allocate the resource; and

 upon receiving the second-request from the client computer, the server computer de-allocating the resource.

21. The method of claim 20, wherein after the server computer has returned the predetermined close instruction, and before the server computer receives the

second request from the client computer, the server computer consecutively sends content pages to the client computer.

22. The method of claim 21, wherein in the step returning a predetermined close instruction, the browser presents the close instruction in a first frame and presents the content pages in a second frame.

23. The method of claim 21, wherein the close instruction prevents selected content pages from being cached by the browser.

24. The method of claim 20, wherein in the step sending a second request, the client computer sends the second request to a predetermined address of the server computer.

25. The method of claim 20, wherein in the step returning a predetermined close instruction, the predetermined close instruction comprises script.

26. The method of claim 20, wherein in the step returning a predetermined close instruction, the script does not lead to a presentation by the browser.

27. A computer program product for HTTP communication between a client computer and a server computer, wherein the client computer includes a browser, the computer program product including program code portions that cause a client

processor in the client computer and a server processor in the server computer to control the communication, the computer program product further comprising:

code portions that cause the client processor to send a first request to the server computer;

code portions that - upon receiving the first request by the server computer - cause the server processor to (i) allocate a resource at the server computer, the resource including an identifier, and (ii) return a predetermined close instruction to the browser, the close instruction carrying the identifier;

code portions that - upon unloading the close instruction from the browser of the client computer - cause the client processor to send a second request to the server computer, the second request carrying the identifier and indicating to de-allocate the resource; and

code portions that - upon receiving the second request from the client computer - cause the server processor to de-allocate the resource.

28. The computer program product of claim 27, wherein the code portions cause the client processor to provide such a close instruction that the browser provides a first frame to present the close instruction in a first frame and provides a second frame to present content pages that the client computer receives from the server computer.

29. The computer program product of claim 27, wherein the code portions cause the client processor to provide such a close instruction that caching of selected content pages by the browser is prevented.

30. The computer program product of claim 27, wherein the code portions cause the client processor to provide such a close instruction that the client computer sends the second request to a predetermined address of the server computer.

31. A computer readable medium storing the program code portions of the computer program product of claim 27 that cause the client processor to operate.

32. A computer readable medium storing the program code portions of the computer program product of claim 27 that cause the server processor to operate.

33. A computer system including a client computer and a server computer, wherein both the client computer and the server computer use HTTP for communication and the client computer uses an HTTP-browser, the computer system characterized in that:

the client computer sends a first request to the server computer;

the server computer upon receiving the first request (i) allocates a resource, the resource including an identifier, and (ii) returns a predetermined close instruction to the browser of the client computer, the close instruction carrying the identifier;

the client computer, upon unloading the close instruction from the browser, sends a second request to the server computer, the second request carrying the identifier and indicating to de-allocate the resource; and

the server computer, upon receiving the second request from the client computer, de-allocates the resource.

34. The computer system of claim 33, wherein the client computer presents the close instruction in a first frame and presents the content pages in a second frame.

35. The computer system of claim 33, wherein the server computer provides the close instruction such that in the client computer the close instruction prevents selected content pages from being cached by the browser.

36. A method for communication between a client computer and a server computer, both computers using the hypertext transfer protocol (HTTP) and the client computer using an HTTP-browser, the method comprising:

 sending a request from the client computer to the server computer;

 upon receiving the request, the server computer:

 allocating a resource at the server computer, the resource including an identifier and a time-out period (T),

 returning a close instruction to the client computer, the close instruction including the time-out period (T) and the identifier,

 measuring the time (t) during which communication between the client computer and the server computer is idle, and

 de-allocating the resource when the measured time (t) reaches the time-out period (T); and

 upon receiving the close instruction the client computer:

 measuring the time (t) during which the communication between the client computer and the server computer is idle, and

displaying a warning to the user if the measured time (t) reaches a predetermined fraction (T/X) of the time-out period (T).

37. A computer program product for controlling HTTP-communication between a client computer and a server computer, wherein the client computer has a browser, the computer program product including a client program portion to control a client processor and a server program portion to control a server processor,

wherein the client program product portion causes the client processor to send a request from the client computer to the server computer;

wherein, upon receiving the request by the server computer, the server program portion causes the server processor to allocate a resource at the server computer, the resource including an identifier and a time-out period (T), to return a close instruction to the client computer, the close instruction including the time-out period (T) and the identifier to measure the time (t) during which communication between the client computer and the server computer is idle, and to de-allocate the resource when the measured time (t) reaches the time-out period (T); and

wherein, upon receiving the close instruction by the client computer, the client program portion causes the client processor to measure the time (t) during which the communication between the client computer and the server computer is idle, and to display a warning to the user if the measured time (t) reaches a predetermined fraction (T/X) of the time-out period (T).

38. A method for communication between a client computer and a server computer, both computers using the hypertext transfer protocol (HTTP) and the client computer using an HTTP-browser, the method comprising:

 sending a first request from the client computer to the server computer;

 allocating a resource at the server computer, the resource including an identifier;

 returning a predetermined response page to the browser, the response page carrying the identifier and carrying browser instructions;

 as instructed by the response page, periodically sending the second requests by the browser to the server computer, the second requests carrying the identifier; and

 at the server computer, periodically checking the arrival of the second requests with the identifier from the client computer and de-allocating the resource when a predetermined time period (T) has lapsed since the last arrival.---

LAW OFFICES

FINNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

IN THE ABSTRACT:

Please insert the following:

---ABSTRACT OF THE DISCLOSURE

A client computer and a server computer communicate via HTTP. The client computer uses a standard HTTP-browser. Substantially simultaneously with establishing a session by allocating a resource at the server computer, the server computer sends a termination instruction to the browser. The instruction remains unexecuted in the browser during the whole session. In the event that the server computer terminates the session (such as upon unloading the instruction from the browser), the browser causes the server computer to de-allocate the resource.---

REMARKS

The Examiner is respectfully requested to enter the above-indicated amendments to the present application prior to the first official examination.

If there is any fee due in connection with the filing of this amendment, please charge any such fee to our Deposit Account No. 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: May 10, 2002

By: 

C. Gregory Gramenopoulos
Reg. No. 36,532

APPENDIX TO AMENDMENT

Applicants submit the following requested changes to the present application with brackets and underlining for the Examiner's convenience as required by 37 C.F.R. § 1.121(c)(1)(ii). This APPENDIX is not intended to be entered into the application.

IN THE SPECIFICATION

Page 1 paragraph 3

Typically, the user operates a personal computer (referred to as "client computer") that has to communicate with a remote computer (referred to as "server computer"). The client computer has communication software to communicate with the server computer [,] and the server computer has application software to execute a business application.

Page 2 first full paragraph

Using specialized communication software at the client computer is inconvenient. Besides the time that is required to install it, installing the communication software might require the payment of license fees. Also, regular and costly updates are required. There is a tendency to communicate with standard "off-the-shelf" software such as [I]Internet browsers. Browsers are installed in almost every personal computer. The hypertext transfer protocol (HTTP) became the standard communication protocol in the [internet] Internet. However, HTTP does not automatically notify the server computer about a session termination by the client computer.

Page 3 paragraph 2

[As expressed in claim 1] According to an embodiment of the present invention, [relates to] a method is provided for communication between a client computer and a

server computer [in] that both [computers] use HTTP. The client computer uses an HTTP-browser. The method comprises the following steps:

Page 4 paragraph 1

It is an advantage that the close instruction in the browser virtually couples the client computer to the server computer. In the event of unloading, the server computer is notified and is able to release the resource. Further, the browser [is] may be a standard browser that interprets the close instruction but that does not need to be modified.

Page 4 paragraph 3 (bridging to page 5)

[As expressed in claim 8,] According to another embodiment of the present invention, [relates to] a computer program product is provided for HTTP-communication between a client computer and a server computer, wherein the client computer has a browser. The computer program product has program code portions that cause a client processor in the client computer and a server processor in the server computer to control the communication. The computer program product comprises: code portions that cause the client processor to send a first request to the server computer; code portions that - upon receiving the first request by the server computer - cause the server processor to (i) allocate a resource at the server computer, the resource [with] including an identifier, and [to] (ii) return a predetermined close instruction to the browser, the close instruction carrying the identifier; code portions that - upon unloading the close instruction from the browser of the client computer - cause the client processor to send a second request to the server computer, the second request carrying the identifier and indicating to de-allocate the resource; and code portions that - upon receiving the

second request from the client computer - cause the server processor to de-allocate the resource.

Page 5 first full paragraph

Preferably, the code portions cause the client processor to provide such a close instruction that the browser provides a first frame to present the close instruction in a first frame and provides a second frame to present content pages that the client computer receives from the server computer. Preferably, the code portions cause the client processor to provide [such] a close instruction such that the caching of selected content pages by the browser is prevented. Preferably, the code portions cause the client processor to provide such a close instruction so that the client computer sends the second request to a predetermined address of the server computer.

Page 5 paragraph 2

[As expressed in claims 12 and 13,] According to still another embodiment of the present invention, [also relates to separate] computer readable media is provided that separately [stores] stores the program code portions causing the client processor and the server processor to operate.

Page 6 paragraph 1

[As expressed in claim 14,] According to another embodiment of the present invention, [relates to] a computer system is provided [in] that includes a client computer and a server computer, both computers using [use] HTTP for communication and [in that] the client computer [uses] using an HTTP-browser. The client computer sends a first request to the server computer; the server computer (upon receiving the first request) (i) allocates a resource (resource [having] including an identifier), and (ii)

returns a predetermined close instruction to the browser of the client computer (the close instruction carrying the identifier); the client computer (upon unloading the close instruction from the browser) sends a second request to the server computer (the second request carrying the identifier and indicating to de-allocate the resource); and the server computer (upon receiving the second request from the client computer) de-allocates the resource.

Page 6 last paragraph (bridging to page 7)

[As expressed in claim 17,] According to yet another embodiment of the present invention, [relates to] a method is provided for communication between a client computer and a server computer. Both computers use HTTP and the client computer uses an HTTP-browser. The client computer sends a request to the server computer. Upon receiving the request, the server computer: allocates a resource at the server computer (the resource [with] including an identifier and a time-out period), returns a close instruction to the client computer (the close instruction [with] including the time-out period and the identifier), measures the time during [that] which communication between the client and server computers is idle, and de-allocates the resource when the measured time reaches the time-out period. Upon receiving the close instruction, the client computer measures the time during [that] which the communication between the client computer and the server computer is idle, and displays a warning to the user if the measured time reaches a predetermined fraction of the time-out period.

Page 7 paragraph 2 (bridging to page 8)

[As expressed in claim 18 ,] In accordance with another embodiment of the present invention, a computer program product is provided for controlling HTTP-

communication between a client computer and a server computer, wherein the client computer has a browser. The computer program product has a client program portion to control a client processor and a server program portion to control a server processor. The program is characterized in that the client program product portion causes the client processor to send a request from the client computer to the server computer; upon receiving the request by the server computer, the server program portion causes the server processor to allocate a resource at the server computer (resource with identifier and time-out period (T)), to return a close instruction to the client computer (close instruction with time-out period (T) and identifier), to measure the time (t) during [that] which communication between the client computer and the server computer is idle, and to de-allocate the resource when the measured time (t) reaches the time-out period (T); and upon receiving the close instruction by the client computer, the client program portion causes the client processor to measure the time (t) during [that] which the communication between the client computer and the server computer is idle, and to display a warning to the user if the measured time (t) reaches a predetermined fraction (T/X) of the time-out period (T).

Page 8 first full paragraph

[As expressed in claim 19,] According to still another embodiment of the present invention, [is described] a method is provided for communication between a client computer and a server computer, (HTTP, client computer with HTTP-browser). [the] The method comprises: sending a first request from the client computer to the server computer; allocating a resource at the server computer, the resource [with] including an identifier; returning a predetermined response page to the browser, the response page

carrying the identifier and carrying browser instructions; as instructed by the response page, periodically sending the second requests by the browser to the server computer, the second requests by the browser to the server computer, the second requests carrying the identifier; and at the server computer, periodically checking the arrival of the second requests with the identifier from the client computer and de-allocating the resource [in case] when a predetermined time period (T) has lapsed since the last arrival.

Page 9 last paragraph (bridging to page 10)

Computers 900-902 are coupled via inter-computer network 990. Computer 900 comprises 910, memory 920, bus 930, and, optionally, input device 940 and output device 950 (I/O devices, user interface 960). As illustrated, features of the invention may be implemented by computer program product 100 (CPP), program carrier 970 and program signal 980, collectively "program".

Page 14 paragraph 2

FIG. 2 illustrates a simplified block diagram of system 999 [in that] including a client computer 900 and a server computer 901 that communicate with each other via network 990 (branch 990-1). Both computers 900 and 901 use the hypertext transfer protocol (HTTP)[;] , and client computer 900 further uses HTTP-browser 210. Browser 210 is the program to locate content pages (e.g., in computer 901) and to display the content pages (presentation 21, cf. FIG. 4). Preferably, browser 210 presents graphics as well as text. Preferably, browser 210 is a Netscape Navigator or a Microsoft Internet Explorer.

Page 15 paragraph 1

During normal operation, browser 210 requests content pages from server computer 901, and server computer 901 responds with content (e.g., HTML pages)[,] . [browser] Browser 210 then causes display 950 to show content pages to the user. An exemplary content page 335 is symbolized by an exclamation mark.

Page 15 paragraph 2

Similar elements such as display, memory, processor, etc. for the other computers are not illustrated for simplicity. Optionally, client computer 900 also communicates with computers 902 and 903 via branches 990-2 and 990-3, respectively[;]. Further, computers 902 and 903 execute applications 402 and 403, respectively. Optionally, the content pages are generated by application computer 902. Likewise, browser 210 requests pages from application computer 902 and application computer 902 responds with application pages.

Page 15 paragraph 3

Preferably, application 403 assists the user to identify application 401 and 402 out of a plurality of applications that are available in the overall network. Such assistance applications are commercially available from SAP Aktiengesellschaft, Walldorf (Baden), Germany under the name "workplace". For convenience, computer 903 is therefore referred to as "workplace computer". Further illustrated elements are: resource 340 (used temporarily), requests 230, 240, instruction 360 and ID 350 (transmitted via the network). The [function] functions of these elements [is] are explained [next :] below.

Page 15 paragraph 4

FIG. 3 illustrates a simplified flow chart diagram of method 500 of the present invention. Method 500 is a method of communication between client computer 900 and server computer 901. Both computers 900, 901 use the hypertext transfer protocol (HTTP) [:] and client computer 900 uses HTTP-browser 210.

Page 17 paragraph first full paragraph

While [The] the present invention is described in connection with a single resource 340, a single session and a single identifier 350, persons of skill in the art are able to implement two or more sessions in parallel.

Page 18 paragraph first full paragraph

It is also possible - although not required for the present invention - that the user terminates the session explicitly, for example, by operating a functional button such as an "abort session" button (cf. 217, 218 or 219 in FIG. 4 or similar ones).

Page 19 paragraph 2

FIG. 4 illustrates display 950 of client computer 900 [for that] with which HTTP-browser 210 (cf. FIG. 2) generates browser presentation 211. Browser presentations are well known in the art[;]. [in] in the example of FIG. 4, presentation 211 has back button 213, address field 214, [and] close button 219, warning 205 and a display area for showing frames.

Page 19 paragraph 5 (bridging to page 20)

Splitting the display screen into frames 215, 216 is convenient for the user[;]. Frame 215 informs that session management (cf. method 500) is active, for example, by informing that the user can now access application 401 (e.g., "APPLICATION READY").

Page 22 paragraph 2

Code section 2 defines a global string to store second request 240 (referred to as "termination URL") for the session in child frame 216. In the example, request 240 does not have any other content[:]. It is [it] sufficient to target second request 240 by identifier 350 to an address in server computer 901 that is reserved for de-allocating resource 340 (for example, http://network-990.server-computer-901.application-401.resoure-340").

Page 22 last paragraph (bridging to page 23)

"This is first frame 215. Client computer 900 with browser 210 had been sending 520 first request 230 (e.g., by URL http://network-990/server-computer-901/application-401) to server computer 901. Upon receiving 530 first request 230, server computer 901 had been allocating 531 resource 340 with identifier 350 and had been returning 532 predetermined close instruction 360 in the form of the present HTML-document "instruction-360.htm" with identifier "340". Close instruction 360 carries identifier 350 ("340"). HTML-document "instruction-360.htm" comprises a termination command in the program section 6 with "onunload". The session has now started."

Page 23 paragraph 5

The advanced session management feature (c) "time" is explained[:] below.

Page 25 first full paragraph

The advanced session management feature (d) "cache prevention" is explained[:] further below.

Page 25 last paragraph

As long as application 401 in server computer 901 is a read-only type application and content pages 335 substantially remain unchanged, displaying cached pages is

convenient. In some instances, especially for business applications, intermediate results stored in resource 340 at server computer 901 diverge from temporarily cached content pages 335(t-1) on client computer 900. This is no longer convenient and might cause serious problems[:].

Page 26 paragraph 3

The following describes an optional implementation [(e)] of distributed session management[:].

Page 26 last paragraph (bridging to page 27)

FIG. 9 illustrates a simplified diagram of computer system 999 for further optional method implementations. Illustrated are computers 900, 901 and 903. Applications are distinguished into main (M) and starter (S) applications, addressed by equal domain names. By providing close instructions, the starter applications (S) provide session management functionality according to the present invention (methods 500 and 600). The example of FIG. 9 is simplified to application 401[; persons] Persons of skill in the art can use the same scheme for multiple applications (with different domains) in parallel and independent from each other.

Page 28 first full paragraph

The following describes a still further optional embodiment_ [(f):]